Manufacturing Interoperability

Dr. Steven Ray, Chief
Manufacturing Systems Integration Division
National Institute of Standards and Technology

The term "manufacturing interoperability" refers to the ability to seamlessly share technical and business information throughout an extended manufacturing enterprise. This information, previously shared informally within a company, must now be passed electronically and error-free to suppliers and customers around the world. Disparate corporate and national cultures make this task all the more difficult, further underscoring the need for clear and unambiguous definitions of terms. The penalty paid by industry for the lack of a strong interoperability infrastructure has been quantified in a 1999 study commissioned by the National Institute of Standards and Technology (NIST),[1] which found that the U.S. automotive sector alone needlessly expends one billion dollars per year due to the lack of interoperability. Up to 50% of these costs are incurred in dealing with data file exchange issues.

There are three principal avenues to achieving interoperability: developing point-to-point translators, mandating the use of proprietary tools and/or formats, or the use of neutral standards.

Point-to-point customized integration can be pursued, often by contracting the services of systems integrators. This approach is expensive in the long run because each pair of systems needs a dedicated solution. Integrating, say, ten systems to one another could require up to 90 one-way translators. If any system provider releases a software upgrade, many of the translators will likely need modification.

In some large supply chains, a dominant OEM (original equipment manufacturer) will mandate that supply chain partners conform to a particular proprietary solution. This has been the practice, for example, in the automotive sector. The problem with this approach is that the interoperability problems are simply pushed lower down the supply chain – they are not eliminated. The first or sub-tier suppliers are forced to purchase and maintain multiple, redundant systems if they want to do business with several major OEMs.

Neutral open standards offer a solution that works well in this environment. Integrating ten systems requires only ten bidirectional translators. Published standards also offer some stability in representation of information, an essential property for long-term data archiving.

The international standard ISO 10303, informally known as STEP – STandard for the Exchange of Product model data, is one example of such an interoperability solution. STEP is actually a family of standards that defines a neutral representation for product data over its entire life cycle.[2] The most widely adopted component, Application Protocol 203[3] is already conservatively estimated to be saving the manufacturing community over $150 million per year in mitigation and avoidance costs, with the figure expected to rise to $700 million by 2010.[4]

But the problem is far from solved. Interoperability standards are used in layers, from the cables and connectors, through the networking standards, to the application or content standards such as STEP. All of these layers must function correctly for interoperability to be achieved. The greatest challenges today

---

[1] Interoperability Cost Analysis of the U.S. Automotive Supply Chain, (Planning Report #99-1), 1999, available at http://www.nist.gov/director/prog-ofc/report99-1.pdf
[2] See http://www.tc184-sc4.org/
[3] ISO 10303-203:1994 Industrial automation systems and integration – Product data representation and exchange -- Part 203: Application protocol: Configuration controlled design
[4] Economic Impact Assessment of the International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries, (Planning Report #02-5), 2002, available at http://www.nist.gov/director/prog-ofc/report02-5.pdf

remain at the top of this stack of standards. Many data interchange standards groups are adopting XML[5] (the eXtensible Markup Language) as the basis for specifying their data content standards. XML has had a tremendously positive impact on the connectivity of systems, but also has more clearly exposed what problems remain. XML is a markup language that can be used to tag collections of data with labels. As part of a standardization activity, communities can agree on the names for these labels. An interoperability problem remains, though, if different people have differing understandings of the meaning of an XML tag. Stated more succinctly, XML standardizes the syntax of data exchange, but was never designed to capture the semantics of the data. This is not necessarily an obstacle for a tightly knit community that operates in a common context. In this situation, the mental associations with a tag are shared and well understood by all. Where this limitation becomes a problem is in moving data from one context to another, for example sending data from a manufacturing context to a financial context. Without explicit, rigorous definitions of terms, misunderstanding is sure to arise.

It was the recognition that we need a better way to capture definitions of terms that resulted in a project led by NIST called the Process Specification Language project.[6] Upon searching for the best conceivable way to capture definitions of terms, the answer was suggested by the philosophy community (which, after all, has been pondering this question for a number of centuries). They suggested the use of first-order logic, which brings the ability to reason over sets of definitions and prove properties of these sets.[7] For example, it becomes straightforward to ensure the consistency of assertions for large sets of definitions by using automated theorem provers – something that is tedious and error-prone using traditional information-modeling techniques.

Ensuring rigorously defined and consistent definitions for data sets is a worthwhile goal in its own right, but the use of formal logic techniques offers even more exciting capabilities. Once a software system is equipped with a logic-based set of definitions (often called an ontology), then it becomes possible for that system to advertise its outputs and desired inputs in a manner that can be manipulated and "understood" by other systems. This begins to move systems beyond the "screen scraping" techniques that are sometimes used today to collect data.

Finally, such a rigorous foundation for data definitions can be the basis for reaching the holy grail of systems integration – that of self-integrating systems. The AMIS project (Automated Methods for Integrating Systems[8]) at NIST builds upon the premise of formal ontologies for defining shared data, and is attempting to automate pieces of the overall integration task. This involves much more than just the data sharing, as one must also capture the interaction between the systems, including what one might call the "choreography" of that interaction.

In summary, manufacturers are becoming ever more reliant upon computer-based systems to house, manipulate, and share their business and technical data. At the same time, manufacturing enterprises are becoming increasingly distributed, requiring the seamless interchange of that data with partners throughout the world. We can no longer afford the inefficiencies of manually ironing out misinterpretations of that data, nor do we have the luxury of sitting down and painstakingly agreeing on how each data interface should be structured. We must look to advanced technological approaches to automate much of this expensive process, much as the "plug and play" approach used for PC backplanes. Such self-describing and self-integrating techniques are not an option, they are a necessity.

---

[5] See http://www.w3.org/XML/

[6] See http://www.nist.gov/psl, and Gruninger, M. and Menzel, C. (2003) Process Specification Language: Principles and Applications, to appear in AI Magazine.

[7] Asher, N. and Vieu, L., Toward a geometry of common sense: a semantics and a complete axiomatization of mereotopology, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 846-852, 1995. Montreal, Canada.

[8] See, for example, www.nist.gov/msidlibrary/doc/AMIS-Concepts.pdf